IFCS 2009
Dresden – 17/03/2009

Visualising a text with a tree cloud

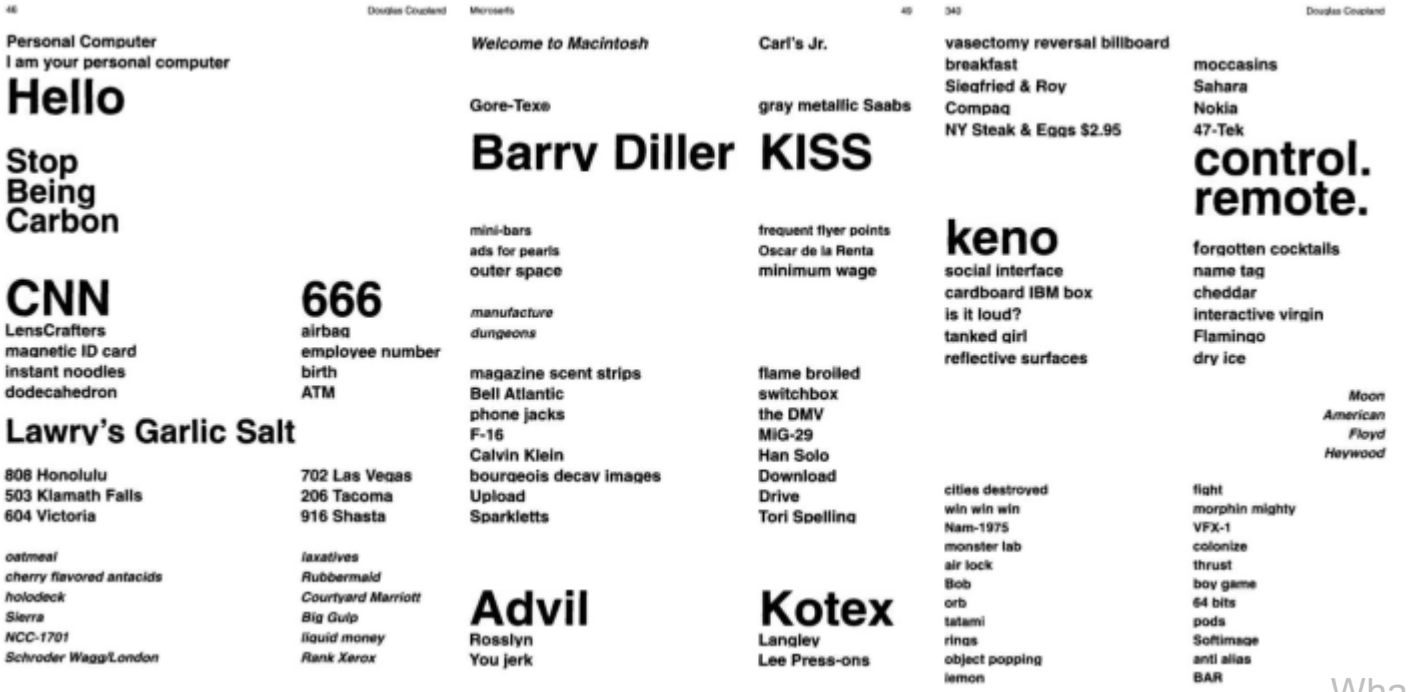
Philippe Gambette, Jean Véronis

Outline

- Tag and word clouds
- Enhanced tag clouds
- Tree clouds
- Construction steps
- Quality control

Tag clouds

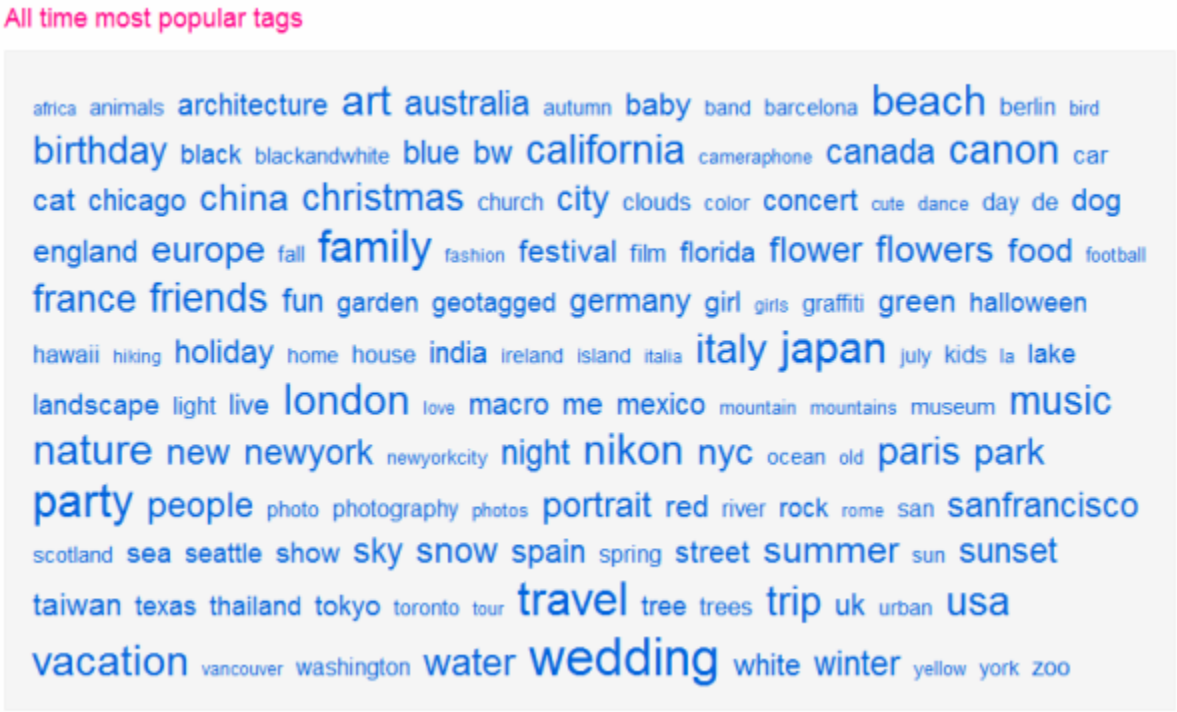
- Built from a set of tags
- Font size related to frequency



What is considered the first tag cloud, from D. Coupland: *Microserfs*, HarperCollins, Toronto, 1995

Tag clouds

- Built from a set of tags
- Font size related to frequency
- Gained popularity with Flickr



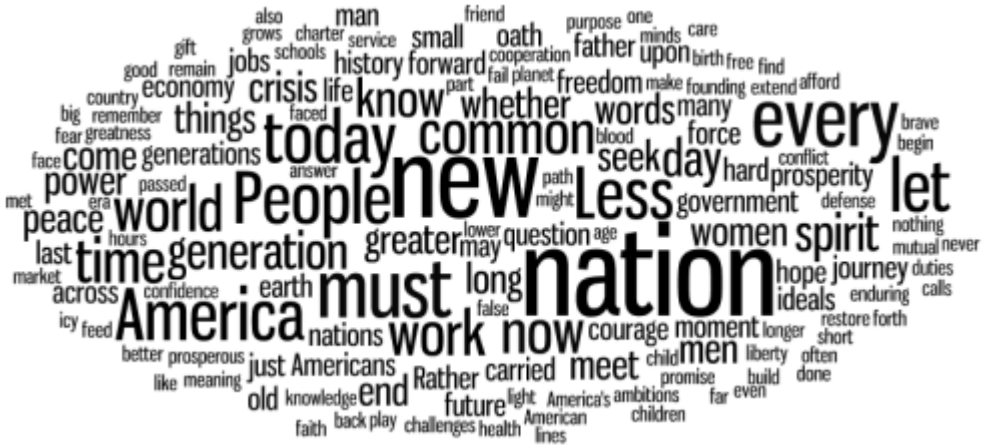
Flickr's *all time most popular tags*

Word clouds

- Built from a set of **words from a text**
- Font size related to frequency
- Gained popularity with Wordle

Word clouds

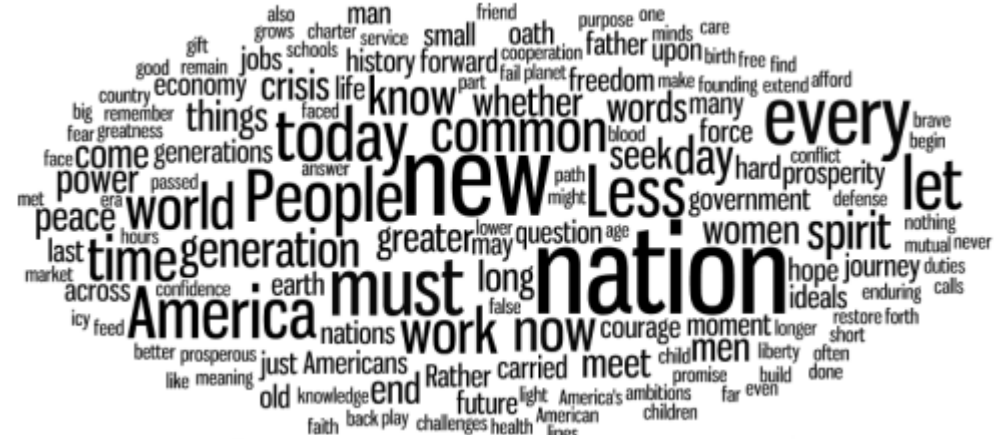
- Built from a set of **words from a text**
- Font size related to frequency
- Gained popularity with Wordle



GoogleImage(obama inaugural address wordle)

Word clouds

- Built from a set of **words from a text**
- Font size related to frequency
- Gained popularity with Wordle



GoogleImage(obama inaugural address wordle)

Word clouds

- Built from a set of **words from a text**
- Font size related to frequency
- Gained popularity with Wordle

GoogleImage(obama inaugural address wordle)

Enhanced tag / word clouds

Add information from the text:

- color intensity to express recency in Amazon
- shared tags in red on del.icio.us
- group together cooccurring tags on the same line

  Hassan-Montero & Herrero-Solana, InScit'06
- optimize blank space and semantic proximity

  Kaser & Lemire, WWW'07
- “topigraphy”: 2D placement according to cooccurrence

  Fujimura, Fujimura, Matsubayashi, Yamada & Okuda, WWW'08

Enhanced tag / word clouds

Add information from the text:

- color intensity to express recency in Amazon
- shared tags in red on del.icio.us
- group together cooccurring tags on the same line

Hassan-Montero & Herrero-Solana, InScit'06

- optimize blank space and semantic proximity

Kaser & Lemire, WWW'07

- “topigraphy”: 2D placement according to cooccurrence

Fujimura, Fujimura, Matsubayashi, Yamada & Okuda, WWW'08

Enhanced tag / word clouds

Add information from the text:

- color intensity to express recency in Amazon
- shared tags in red on del.icio.us
- group together cooccurring tags on the same line

Hassan-Montero & Herrero-Solana, InScit'06

- optimize blank space and semantic proximity

Kaser & Lemire, WWW'07

- “topigraphy”: 2D placement according to cooccurrence

Fujimura, Fujimura, Matsubayashi, Yamada & Okuda, WWW'08

Enhanced tag / word clouds

Add information from the text:

- color intensity to express recency in Amazon
- shared tags in red on del.icio.us
- group together cooccurring tags on the same line

Hassan-Montero & Herrero-Solana, InScit'06

- optimize blank space and semantic proximity

Kaser & Lemire, WWW'07

- “topigraphy”: 2D placement according to cooccurrence

Fujimura, Fujimura, Matsubayashi, Yamada & Okuda, WWW'08

Enhanced tag / word clouds

Add information from the text:

- color intensity to express recency in Amazon
- shared tags in red on del.icio.us
- group together cooccurring tags on the same line

Hassan-Montero & Herrero-Solana, InScit'06

- optimize blank space and semantic proximity

Kaser & Lemire, WWW'07

- “topigraphy”: 2D placement according to cooccurrence

Fujimura, Fujimura, Matsubayashi, Yamada & Okuda, WWW'08

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
sense desamibiguation

Véronis (Hyperlex)

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

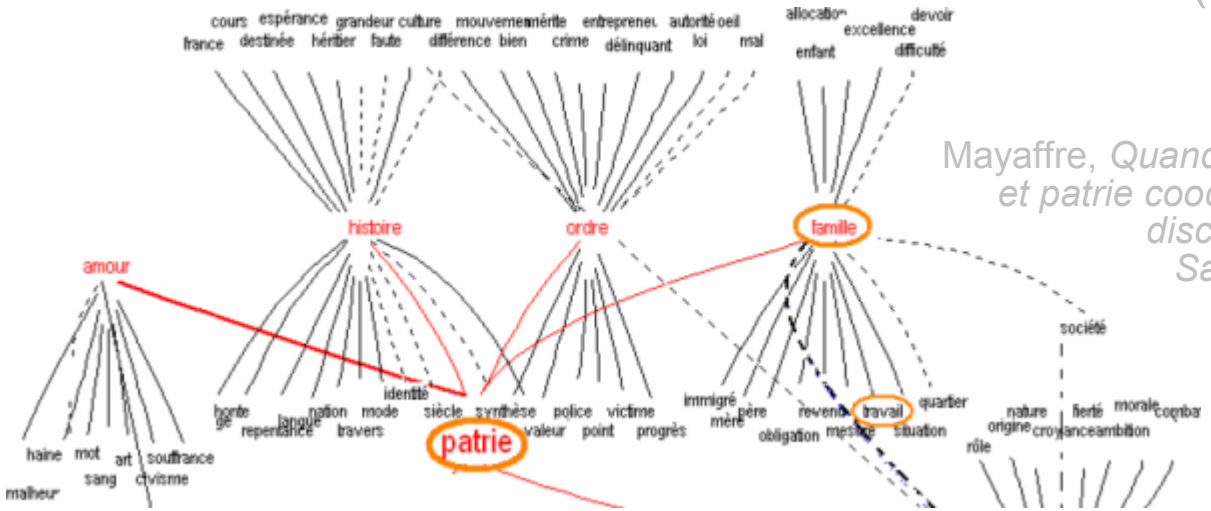
Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
sense desamibiguation

Véronis (Hyperlex)



Mayaffre, *Quand travail, famille, et patrie cooccurrent dans le discours de Nicolas Sarkozy*, JADT'08

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

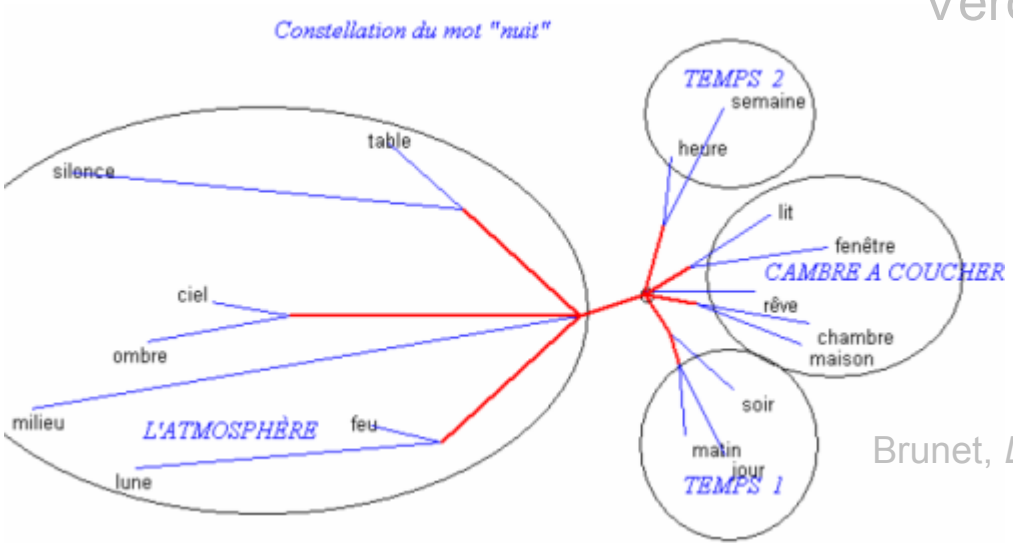
Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
sense desamibiguation

Véronis (Hyperlex)



Brunet, *Les séquences (suite)*, JADT'08

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

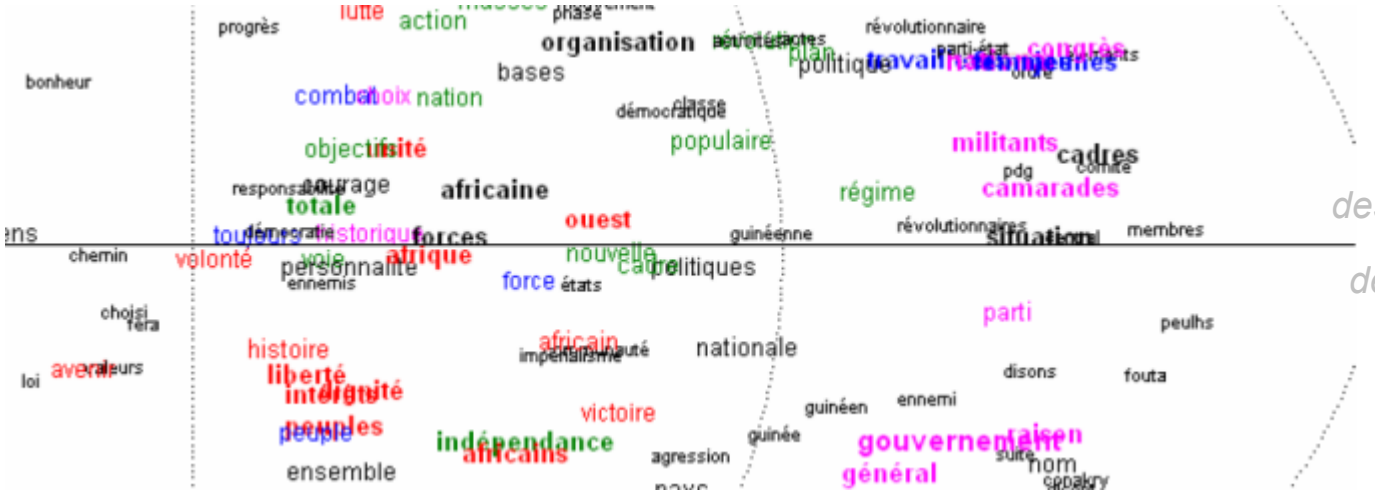
Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
word sense disamibiguation

Véronis (Hyperlex)



Barry, Viprey,
Approche comparative des résultats d'exploration textuelle des discours de deux leaders africains Keita et Touré,
JADT'08

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

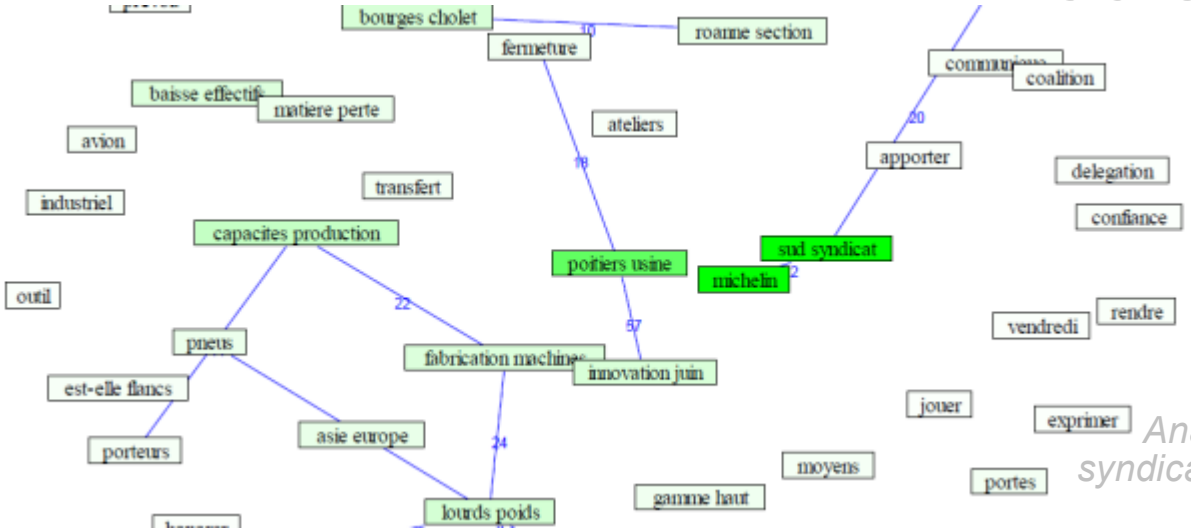
Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
word sense disamibiguation

Véronis (Hyperlex)



Peyrat-Guillard, *Analyse du discours syndical sur l'entreprise*, JADT'08

Extract semantic information from a text

- literature analysis:
philological approach: only consider the text

Brody

- discourse analysis:
tree analysis or cooccurrence graph, geodesic projection

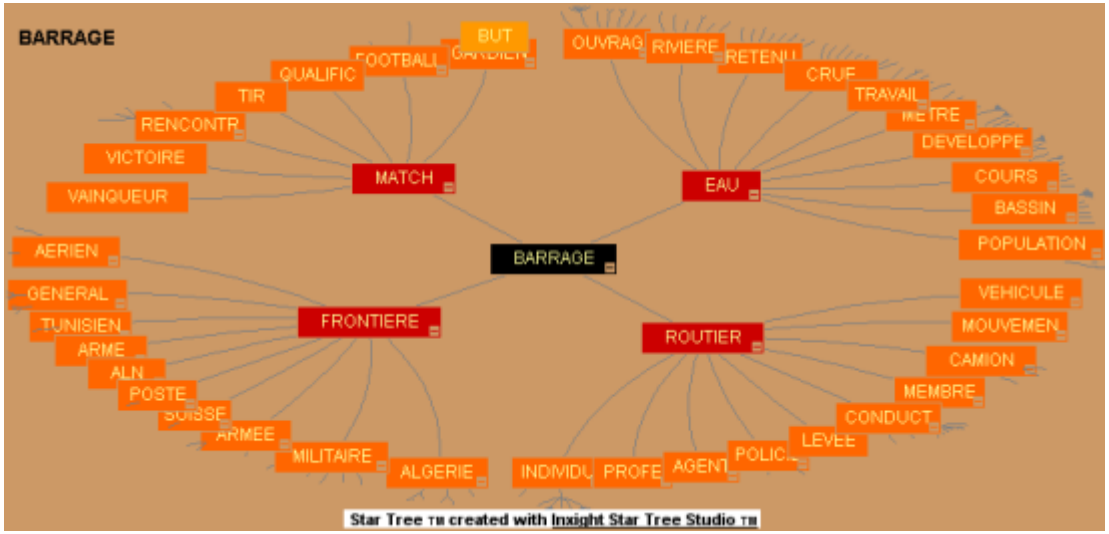
Brunet (Hyperbase), Viprey (Astartex)

- text mining:
semantic graph

Grimmer (Wordmapper)

- natural language processing:
word sense disambiguation

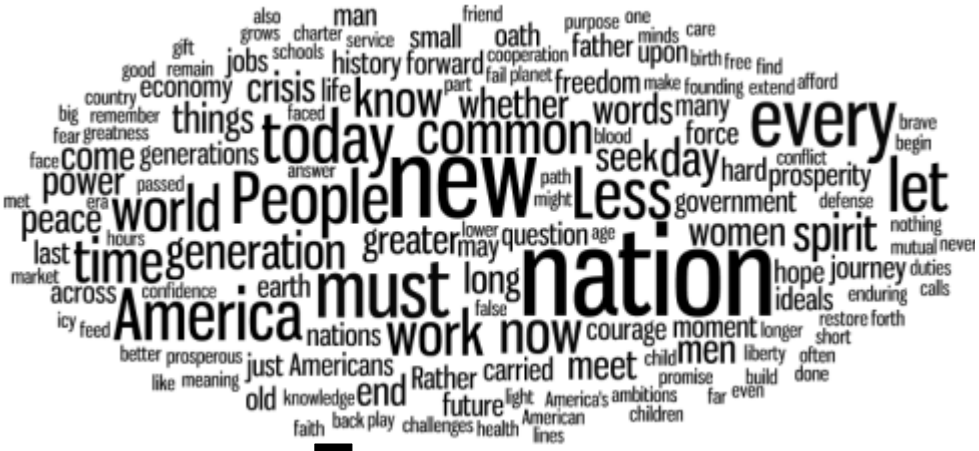
Véronis (Hyperlex)

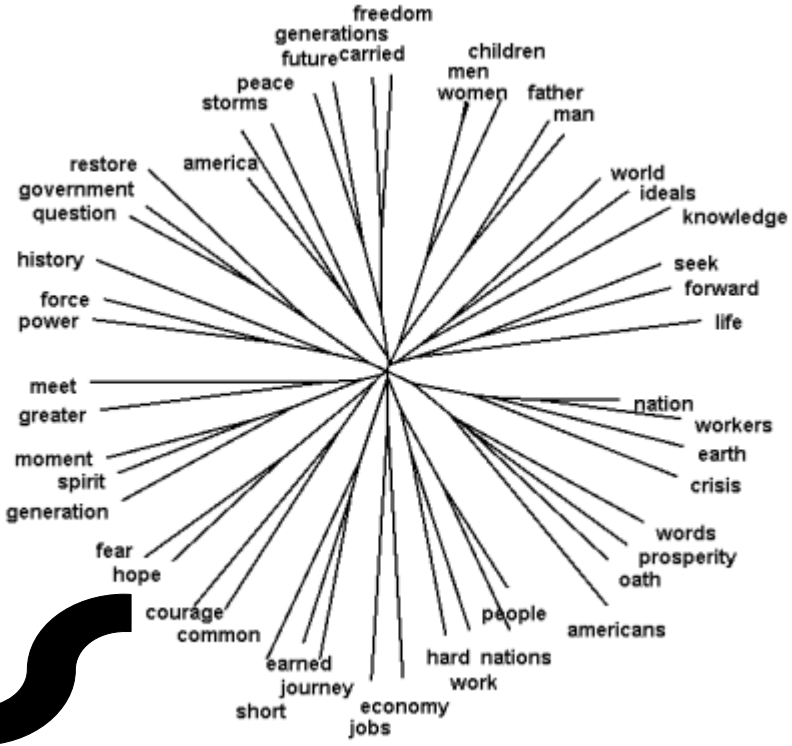


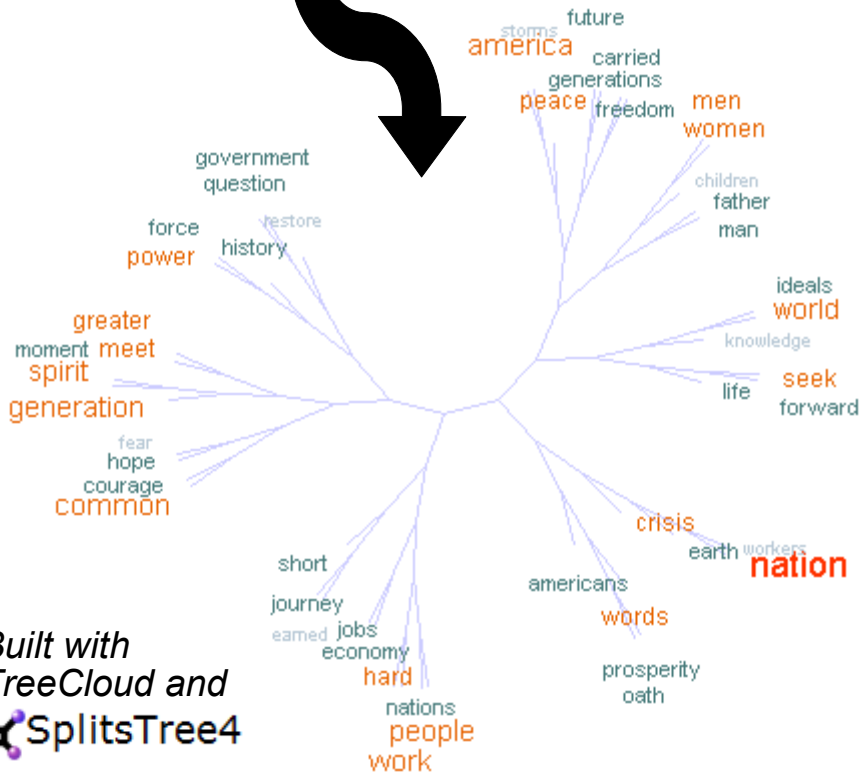
Disambiguation of word “barrage”: dam, play-off, roadblock, police cordon.

Véronis, *HyperLex: Lexical Cartography for Information Retrieval*, 2004

Tag cloud + tree = tree cloud







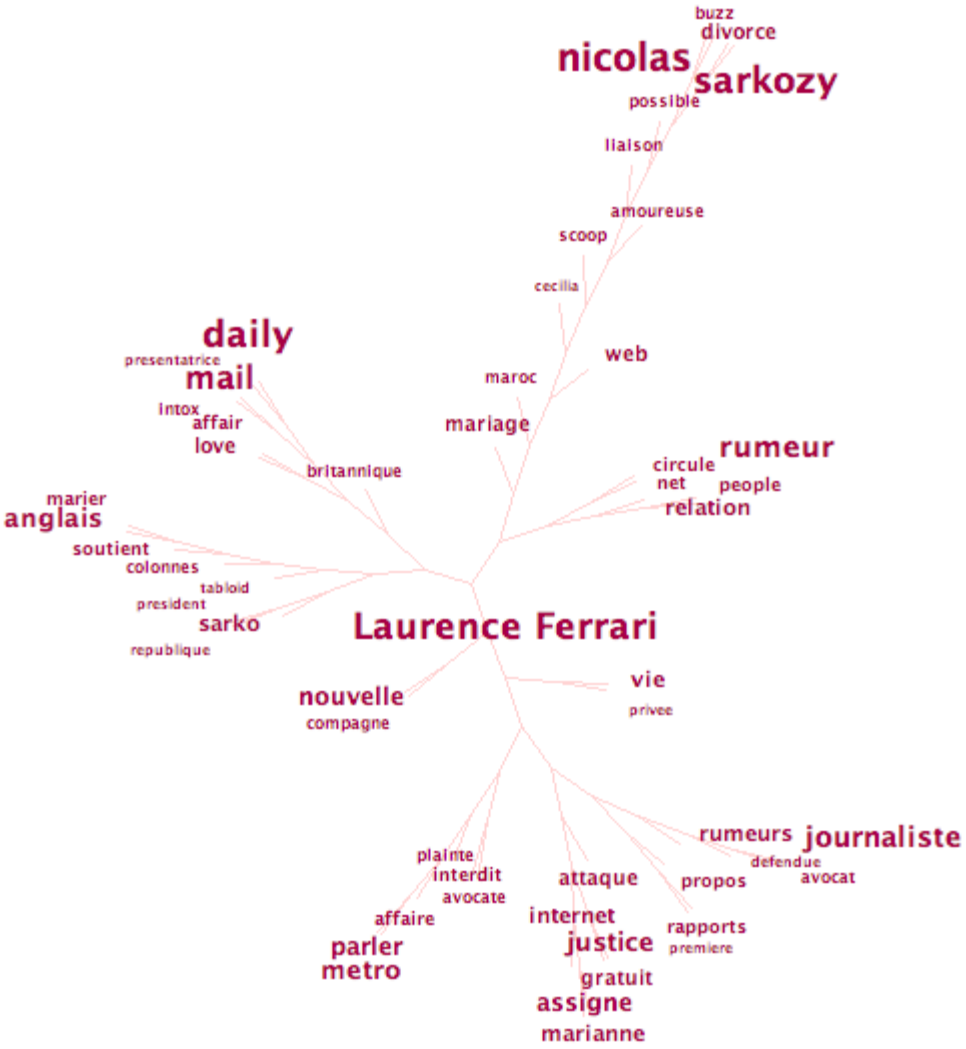
Built with TreeCloud and

SplitsTree4

SplitsTree: Huson 1998, Huson & Bryant 2006

GPL-licensed Treecloud in Python, available at http://www.treecloud.org

The first tree cloud



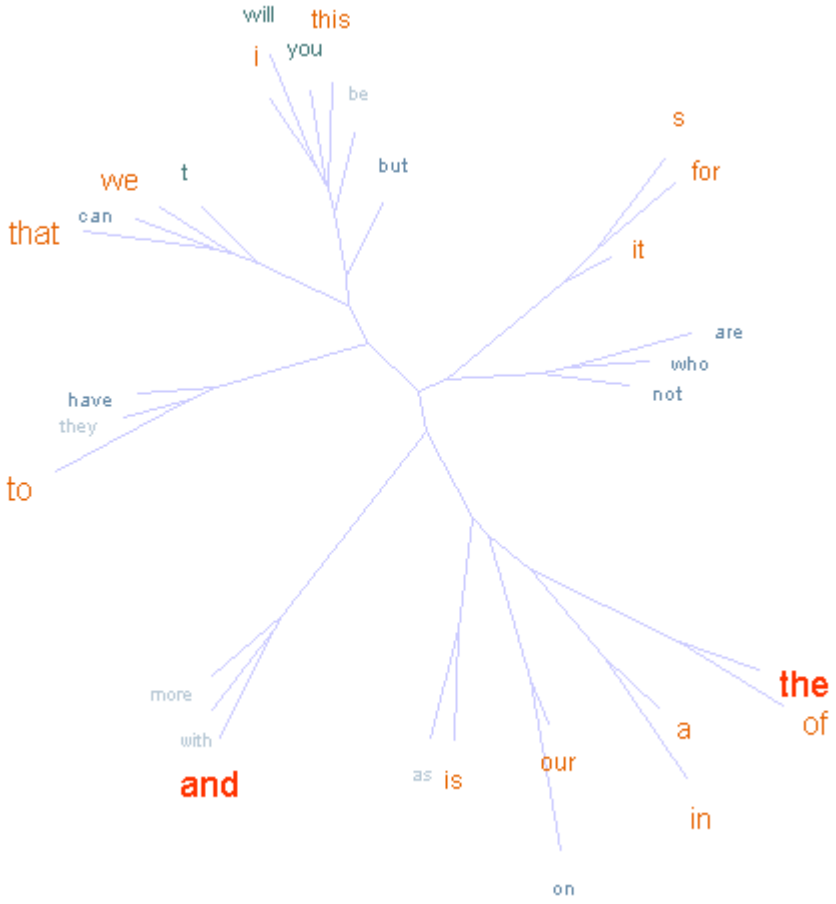
Tree cloud of the blog posts containing “Laurence Ferrari”
from 25/11/2007 to 10/12/2007, by Jean Véronis
http://aixtal.blogspot.com/2007/12/actu-une-ferrari-dans-un-arbre.html

Building a tree cloud – extracting the words

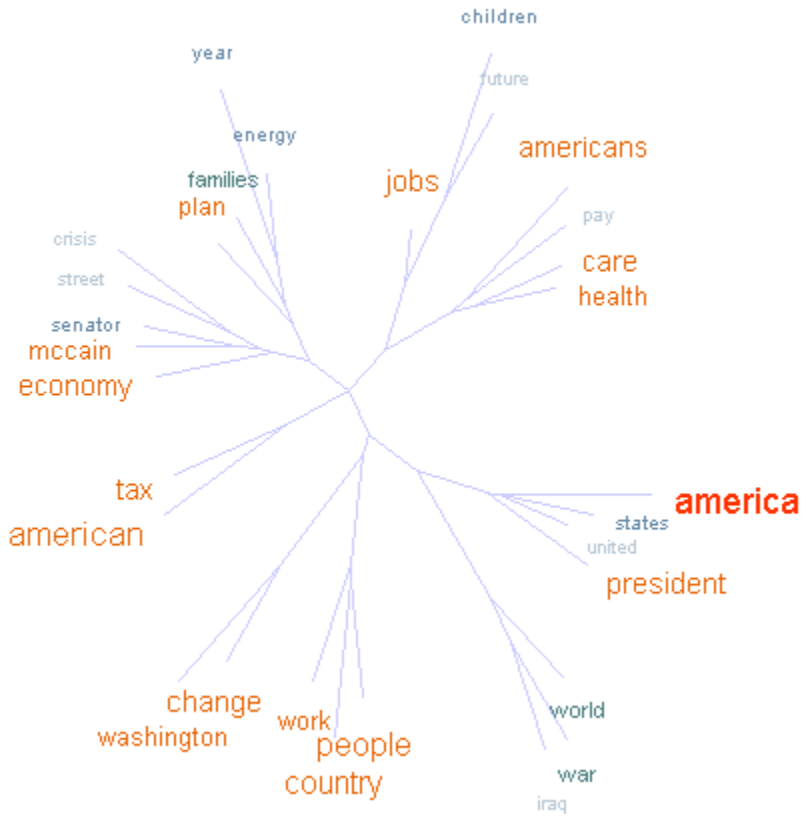
Extract words with frequency:

- stoplist?

without stoplist



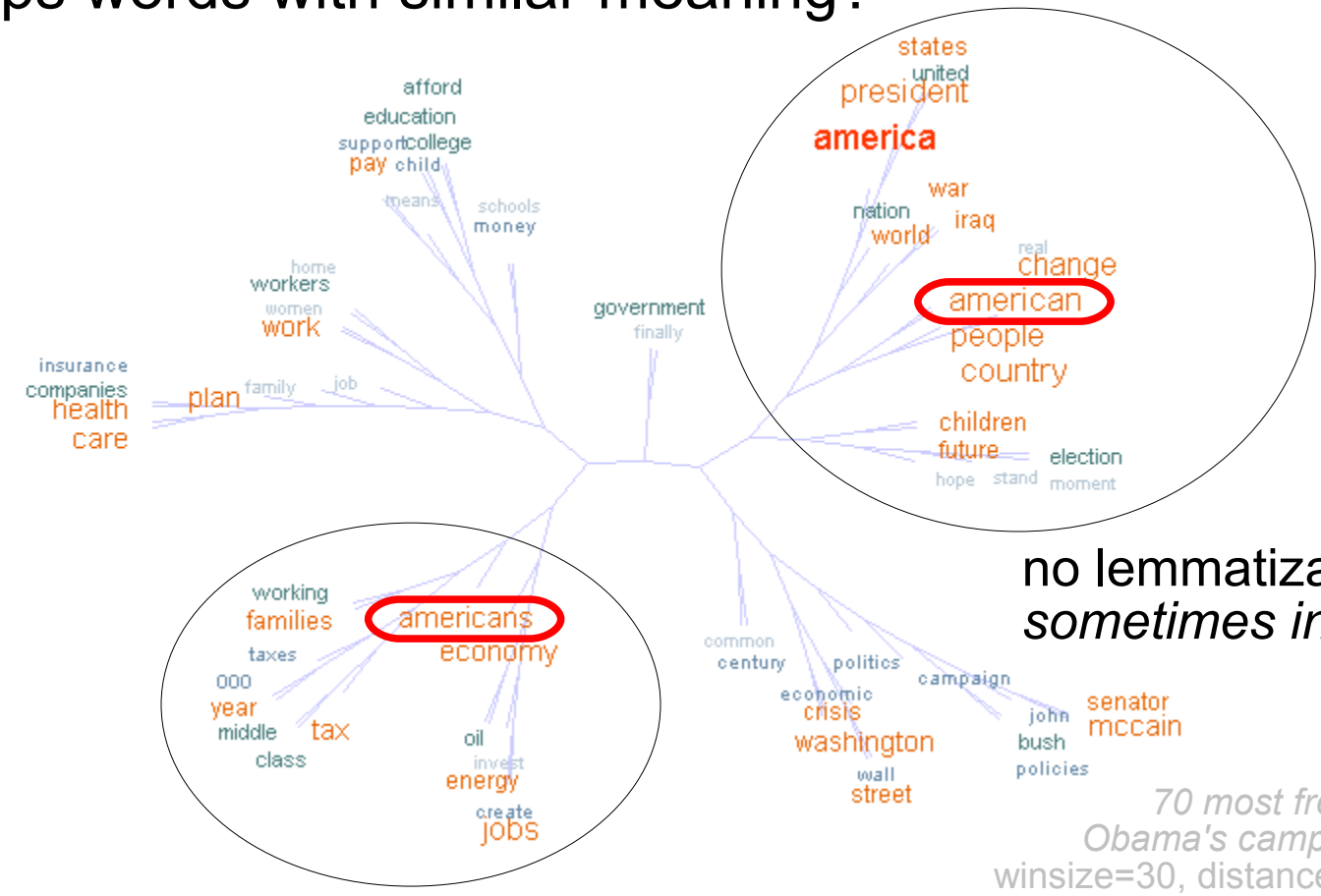
with stoplist



Building a tree cloud – extracting the words

Extract words with frequency:

- lemmatization?
- groups words with similar meaning?



no lemmatization...
sometimes interesting

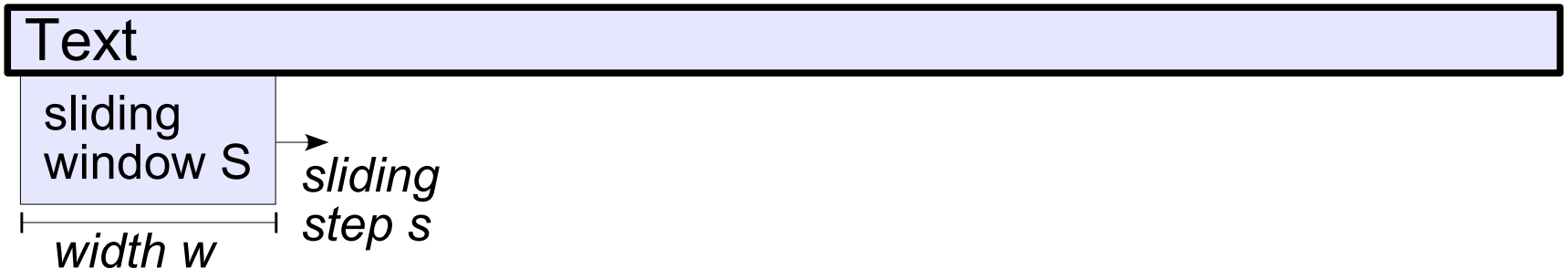
70 most frequent words in Obama's campaign speeches, winsize=30, distance=dice, NJ-tree.

Building a tree cloud – dissimilarity matrix

Many semantic distance formulas based on cooccurrence

Building a tree cloud – dissimilarity matrix

Many semantic distance formulas based on cooccurrence



cooccurrence matrices O_{11}, O_{12}, O_{21}, O_{22}

	$v \in S$	$v \notin S$
$u \in S$	O_{11}	O_{12}
$u \notin S$	O_{21}	O_{22}

semantic dissimilarity matrix

chi squared, mutual information, liddel, dice, jaccard, gmean, hyperlex, minimum sensitivity, odds ratio, zscore, log likelihood, poisson-stirling...

Evert, *Statistics of words cooccurrences*, PhD Thesis, 2005

Building a tree cloud – dissimilarity matrix

Transformations needed on the dissimilarity:

- transform similarity into dissimilarity

- linear normalization for positive matrices to get distances in [0,1]

- affine normalization for matrices with positive or negative numbers, to get distances in [α,1] (for example α=0.1)

Building a tree cloud – tree reconstruction

Many existing methods:

- Neighbor-Joining

Saitou & Nei, 1987

SplitsTree4

- Addtree variants

Barthelemy & Luong, 1987

- Quartet heuristic

Cilibrasi & Vitanyi, 2007

Building a tree cloud – tree decoration

Choice of word sizes:

- computed directly from **frequency** (apply a log!)

or

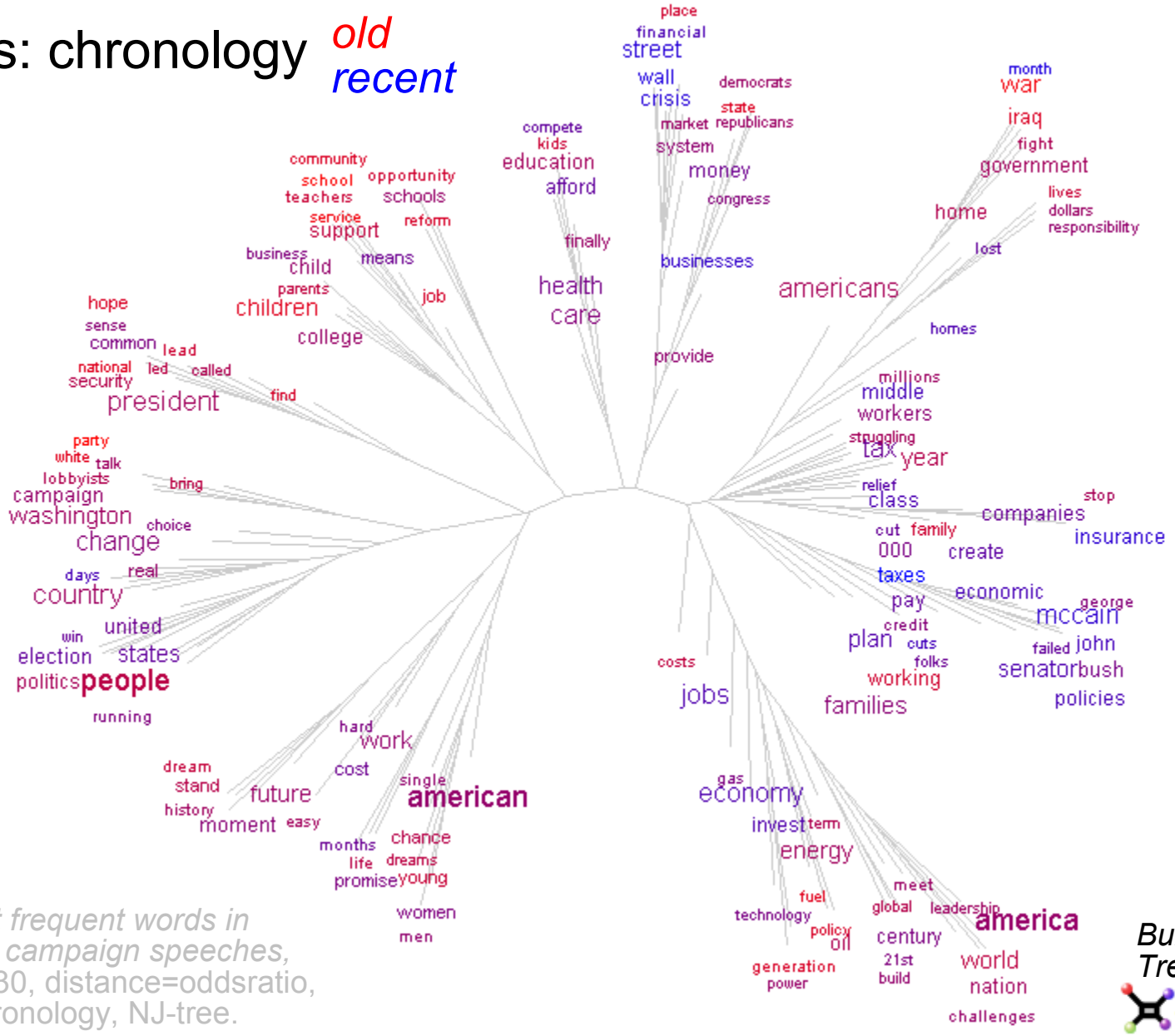
- computed from **frequency ranking** (exponential distribution)

or

- **statistical significance** with respect to a reference corpus

Building a tree cloud – tree decoration

Colors: chronology *old* *recent*



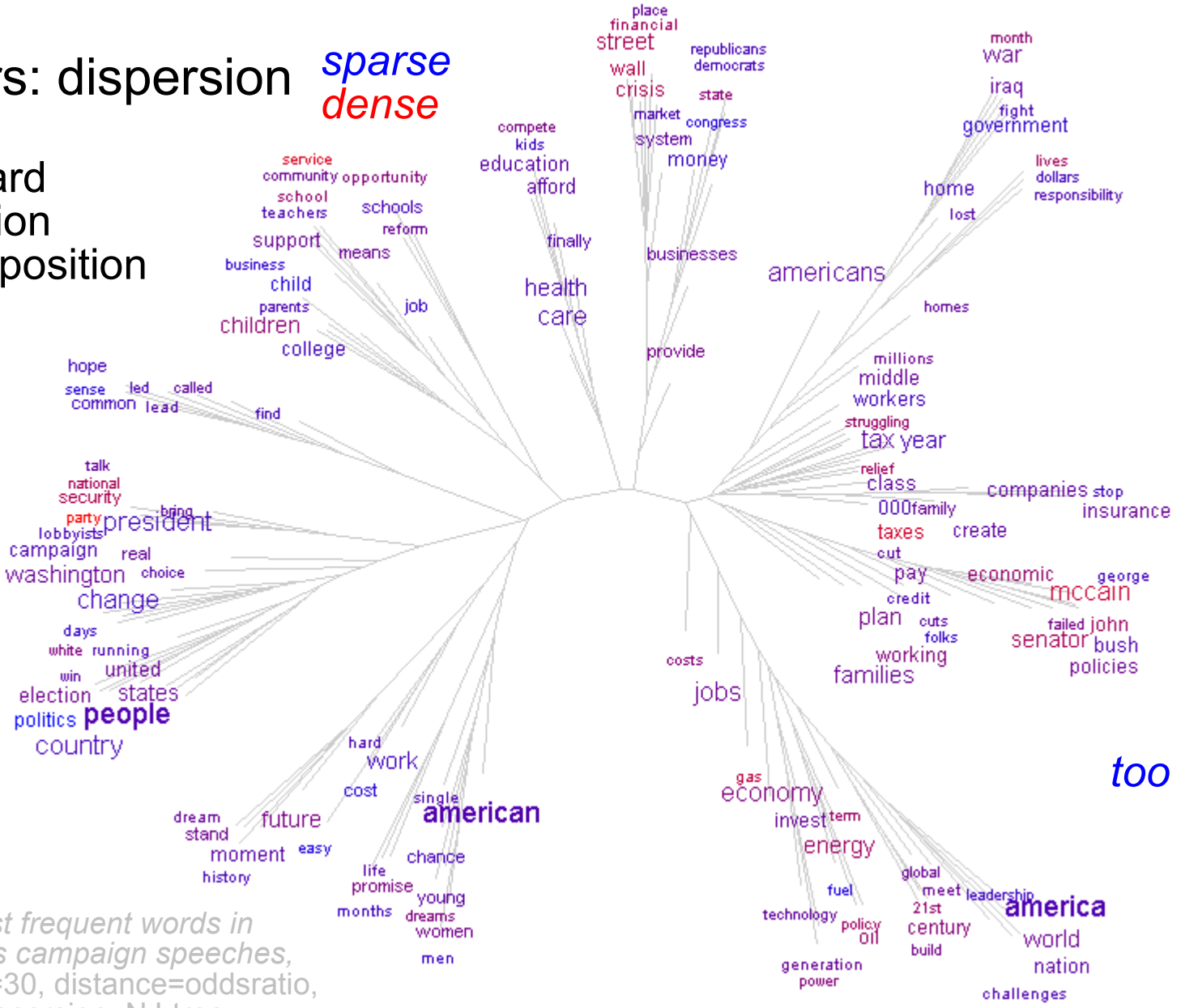
150 most frequent words in Obama's campaign speeches, winsize=30, distance=oddsratio, color=chronology, NJ-tree.

Built with TreeCloud and SplitsTree4

Building a tree cloud – tree decoration

Colors: dispersion *sparse* *dense*

standard deviation of the position



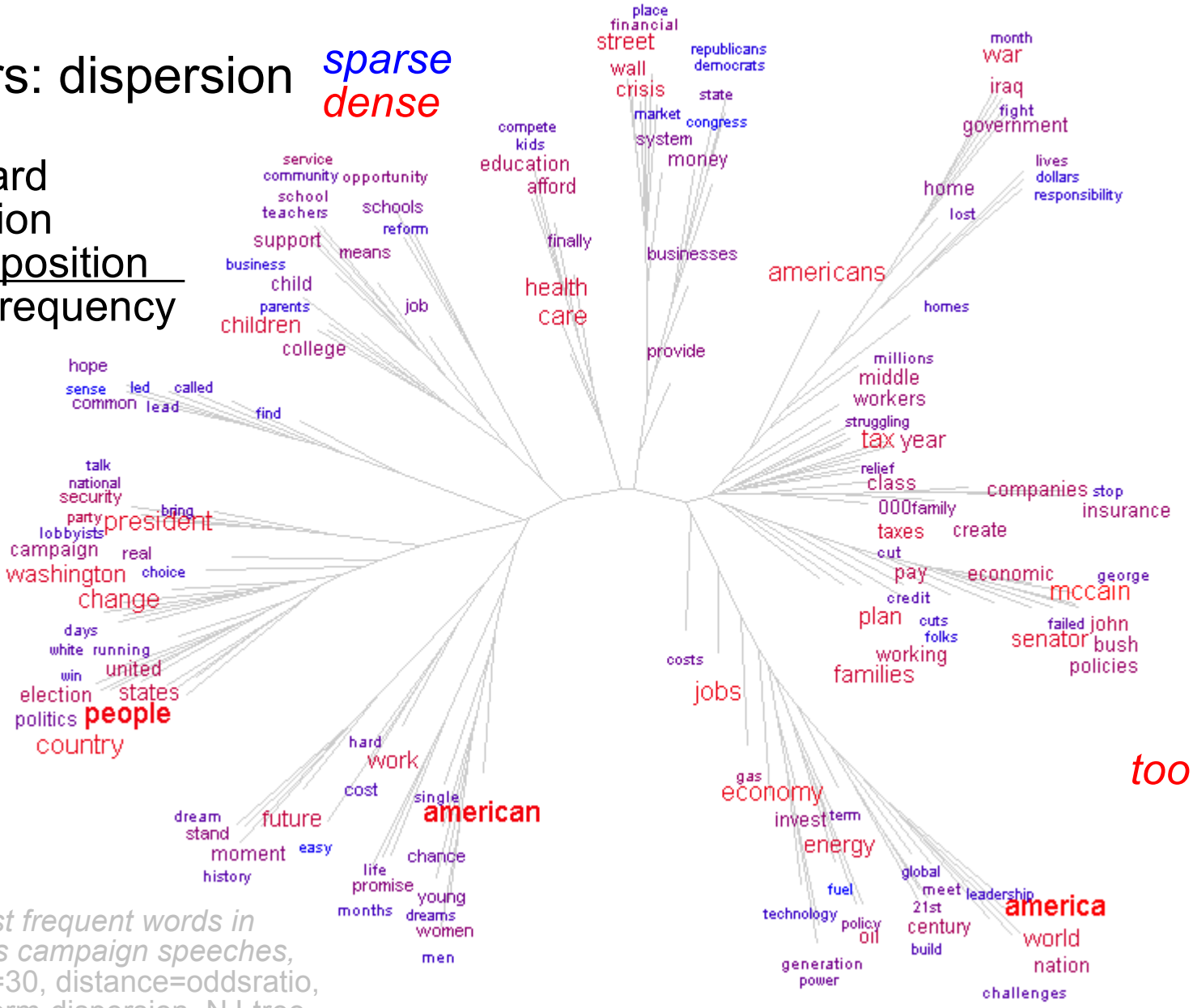
too blue?

150 most frequent words in Obama's campaign speeches, winsize=30, distance=oddsratio, color=dispersion, NJ-tree.

Building a tree cloud – tree decoration

Colors: dispersion *sparse* *dense*

standard deviation of the position / word frequency

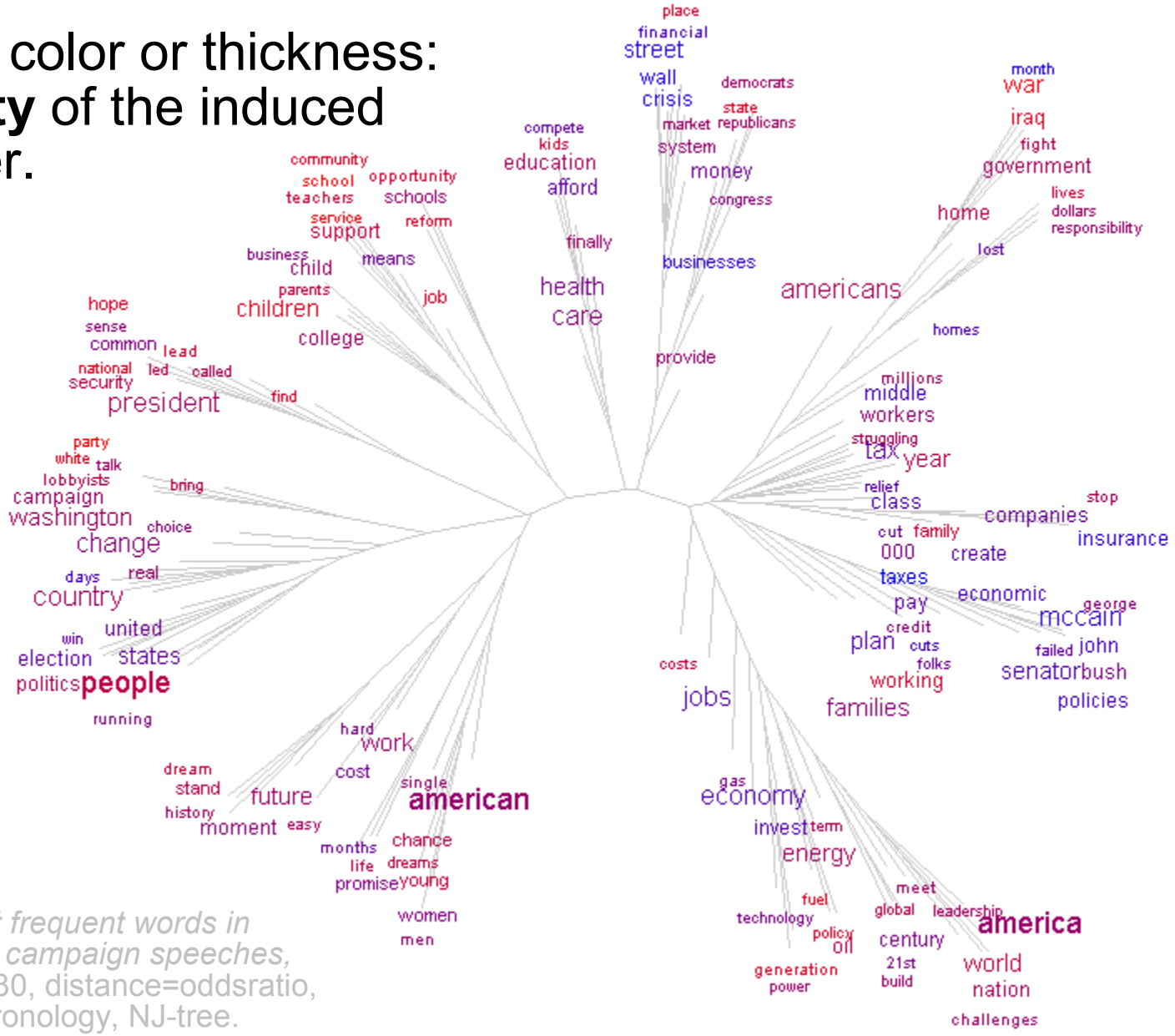


too red?

150 most frequent words in Obama's campaign speeches, winsize=30, distance=oddsratio, color=norm-dispersion, NJ-tree.

Building a tree cloud – tree decoration

Edge color or thickness: **quality** of the induced cluster.



150 most frequent words in Obama's campaign speeches, winsize=30, distance=oddsratio, color=chronology, NJ-tree.

Quality control

Is there an objective quality measure of tree clouds?

Quality control

Is there an objective quality measure of tree clouds?

What is the best method to build a tree cloud from my data?

Quality control

Is there an objective quality measure of tree clouds?

What is the best method to build a tree cloud from my data?

Tree cloud variations if small changes?

➡ **bootstrap** to evaluate:

- **stability of the result**
- **robustness of the method**

Quality control

Is there an objective quality measure of tree clouds?

What is the best method to build a tree cloud from my data?

Tree cloud variations if small changes?

➡ **bootstrap** to evaluate:

- **stability of the result**
- **robustness of the method**

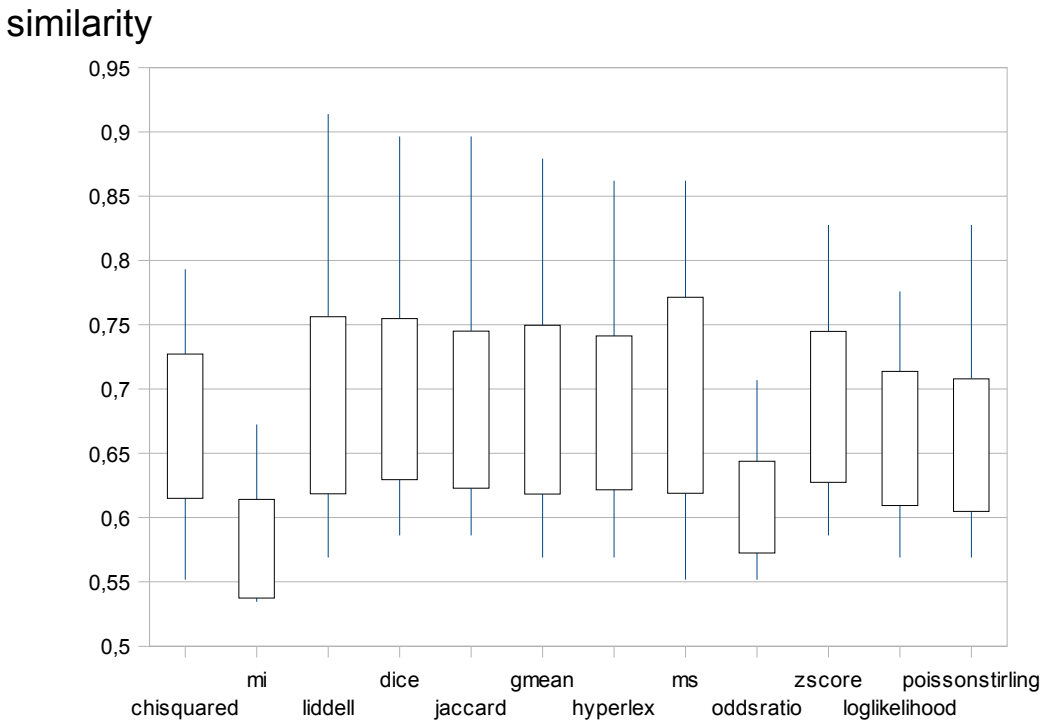
Still, is there a more direct method?

➡ a**rboricity** to show whether the distance matrix **fits with a tree**, which should **imply stability**?

Guénoche & Garreta, 2001
Guénoche & Darlu, 2009

Quality control – bootstrap

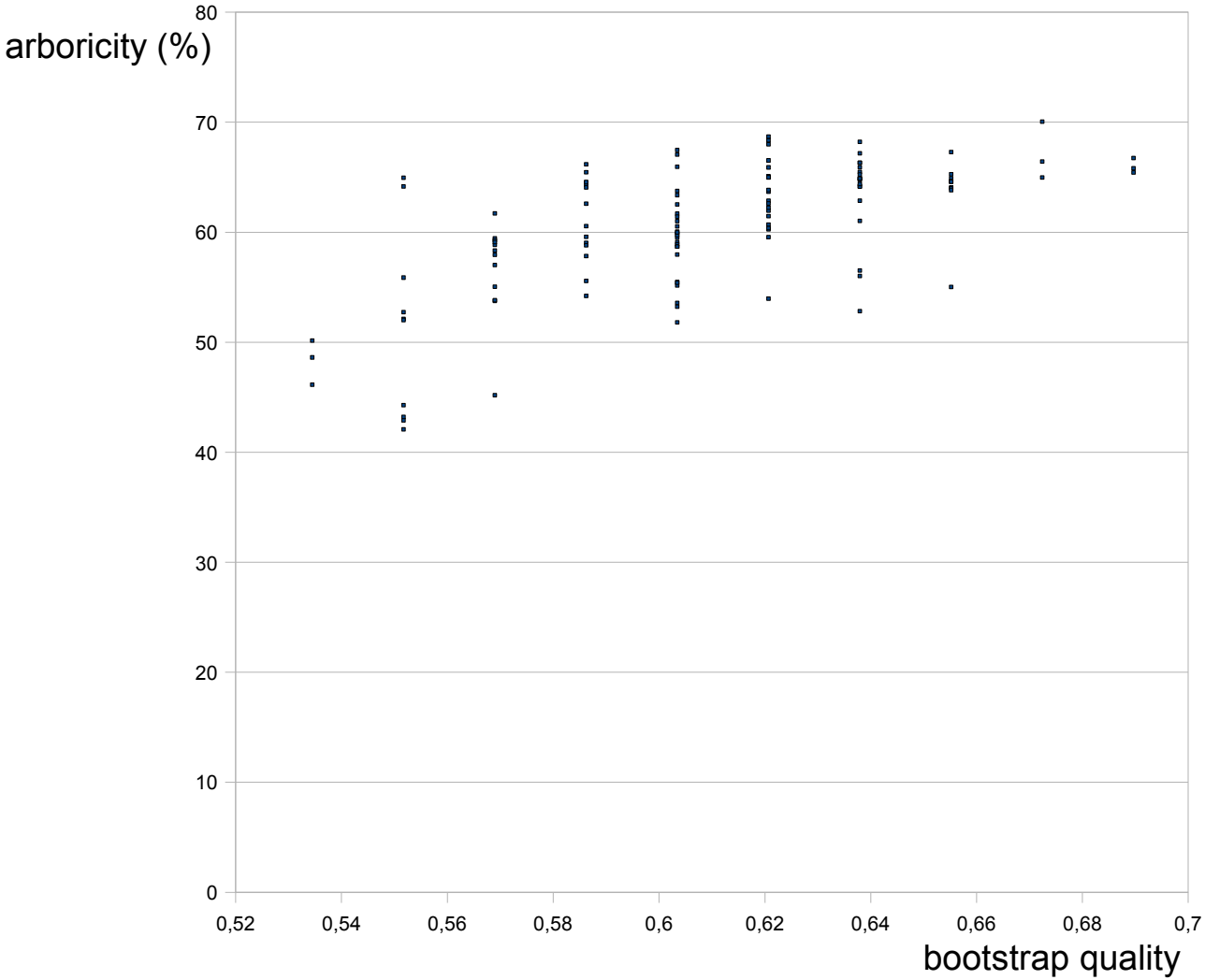
- Randomly delete words with probability 50%.
- Built tree cloud of original text, and altered text.
- Compute similarity of both trees (1-normalized RobinsonFoulds)



4 altered versions of 10 Obama's speeches, 3000 words in average, width=30, NJ-tree.

Quality control – arboricity

Relationship between “bootstrap quality” and arboricity:

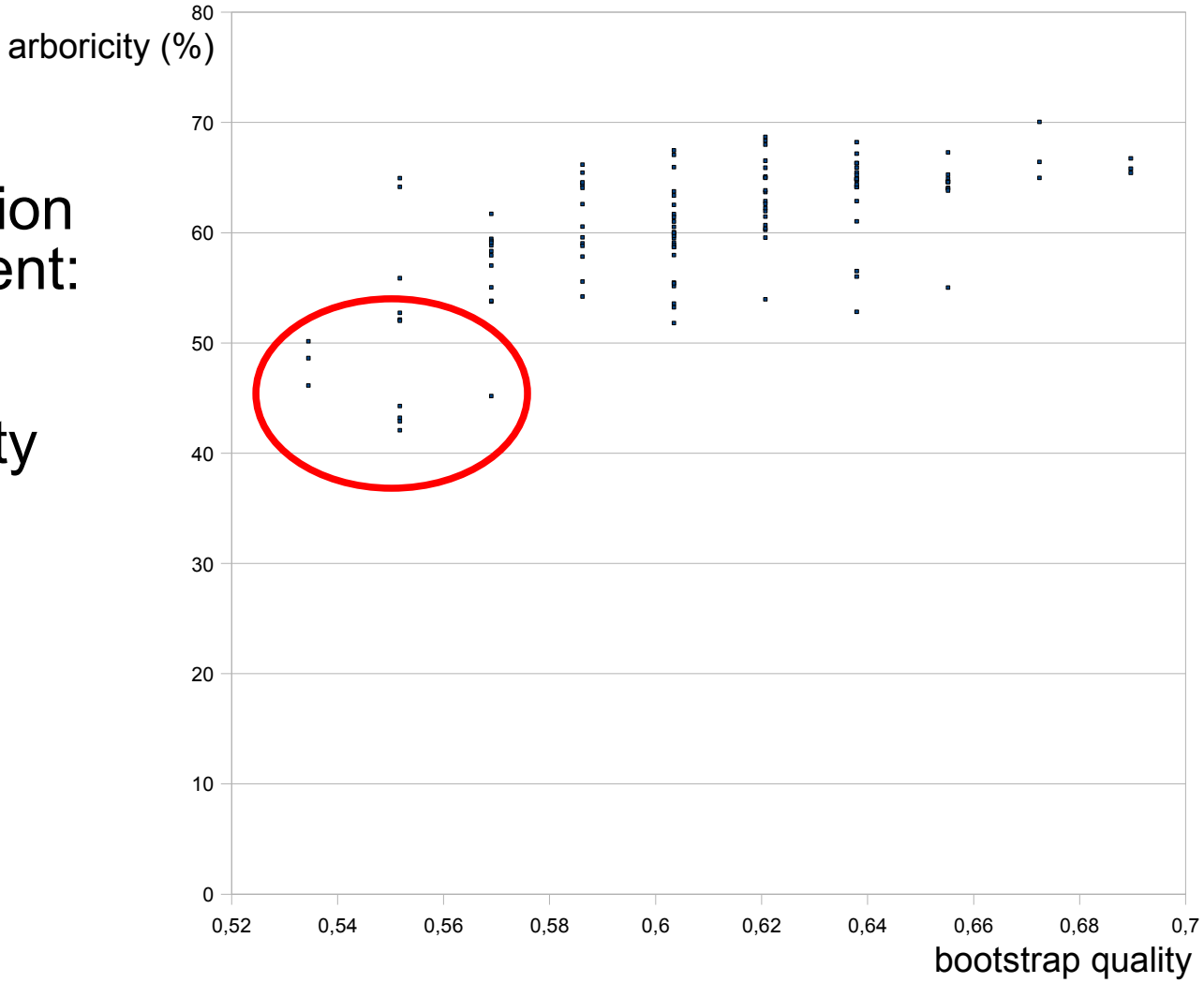


Quality control – arboricity

Relationship between “bootstrap quality” and arboricity:

correlation coefficient: 0.64

arboricity below 50%: danger!



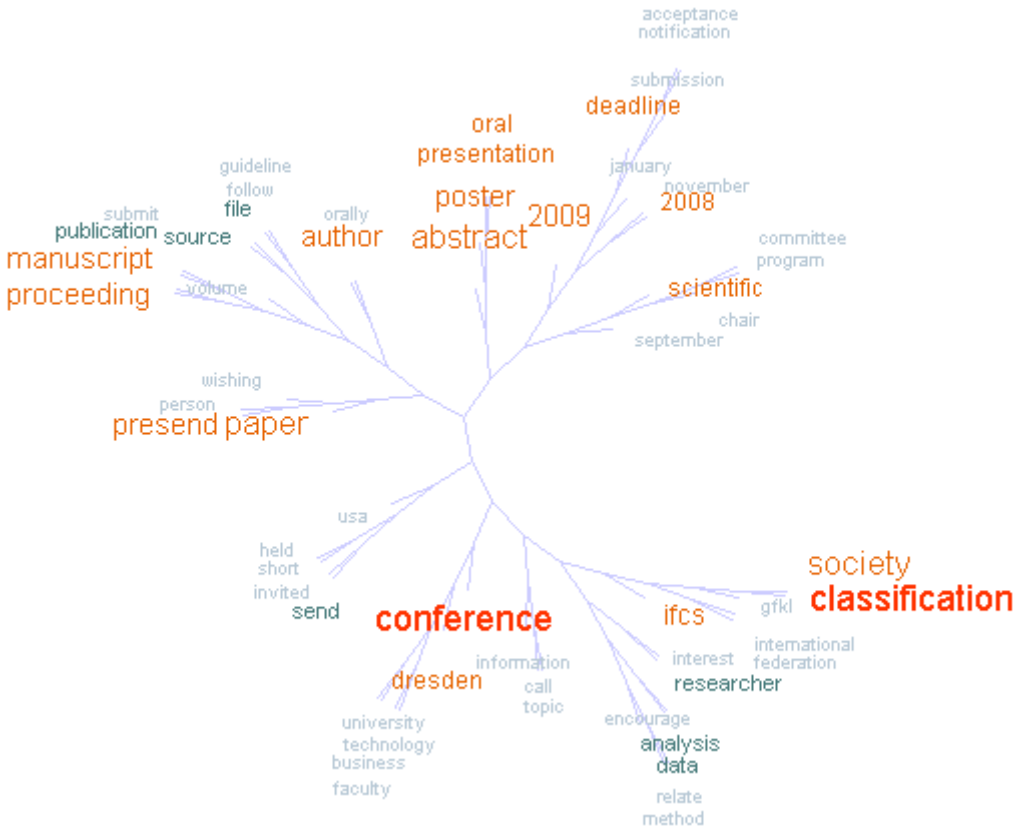
Perspectives

- Make the tool available on a web interface http://www.treecloud.org
- Evaluate tree clouds for discourse analysis
- Build the daily tree cloud of people popular on blogs, with WIKIO Labs

Allemagne

http://labs.wikio.net

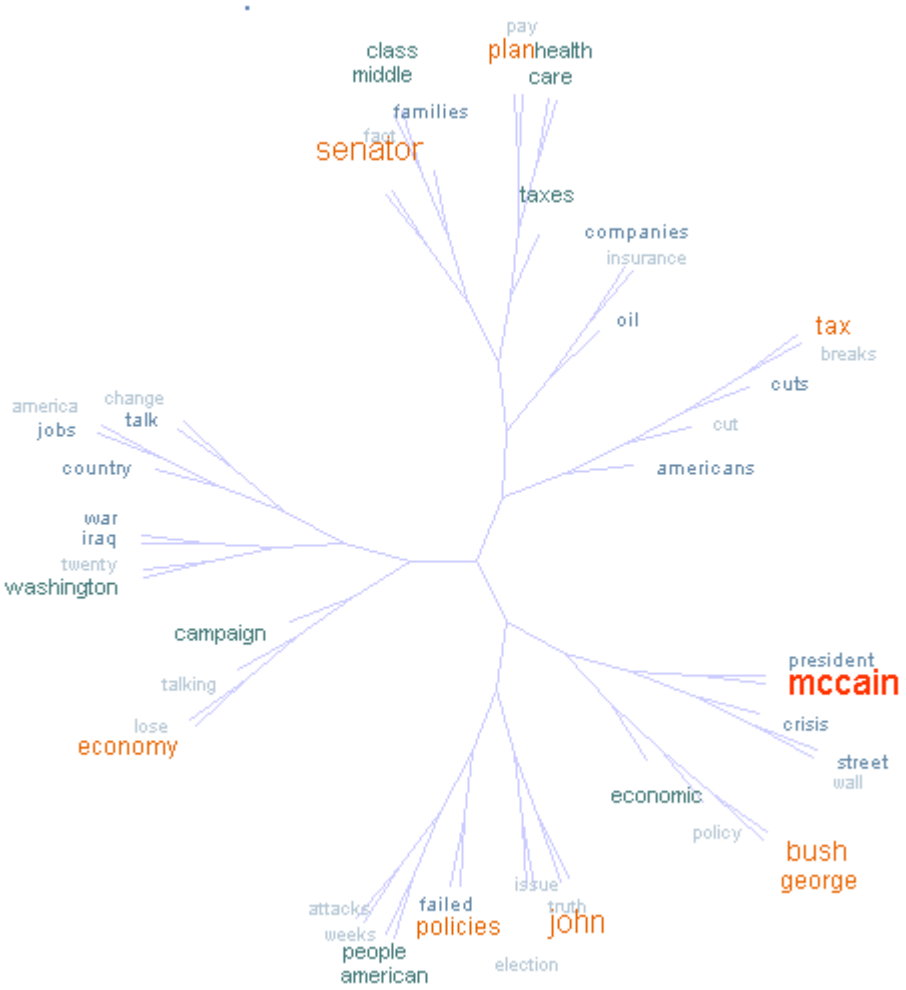
Thank you for your attention!



Tree cloud of the words appearing twice or more in the IFCS 2009 call for paper
lemmatization, width=20, distance=dice, NJ-tree.

http://www.treecloud.org - http://www.splitstree.org

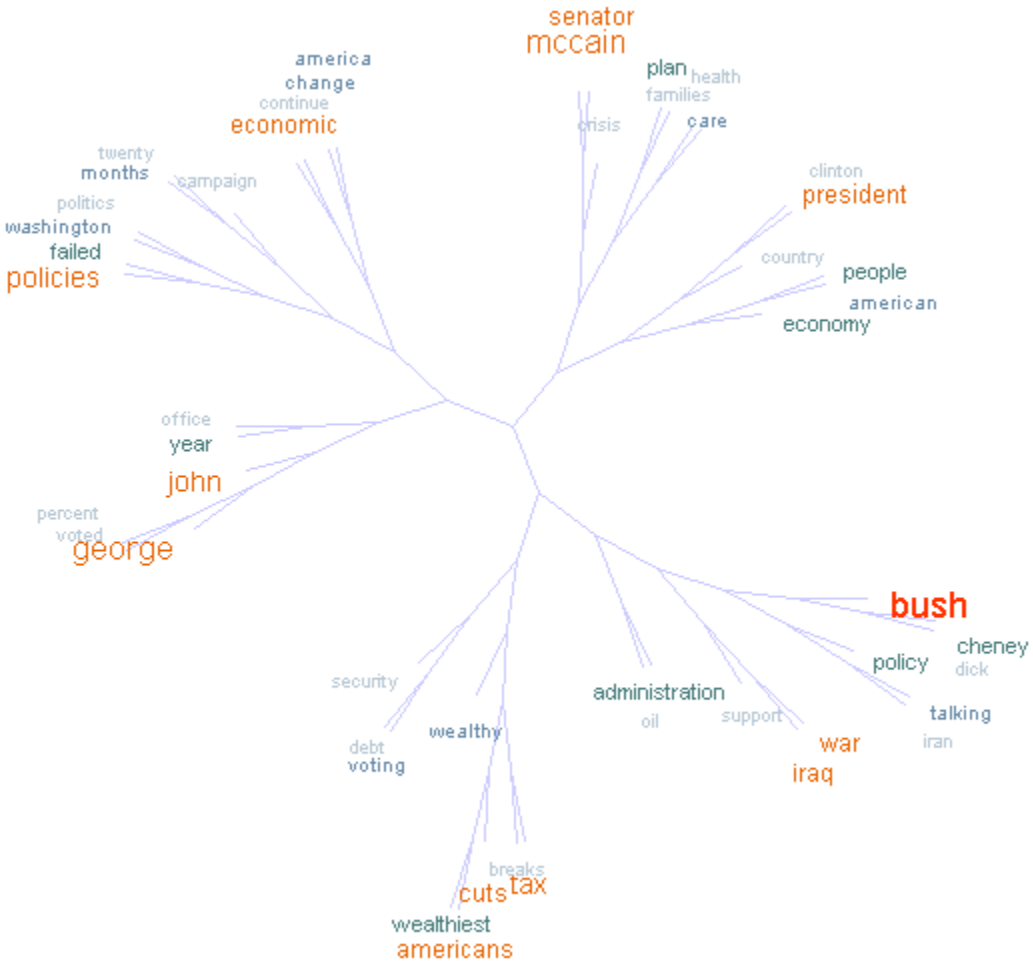
Tree clouds focused on one word



Tree cloud of the neighborhood of "McCain" in Obama's campaign speeches

http://www.treecloud.org - http://www.splitstree.org

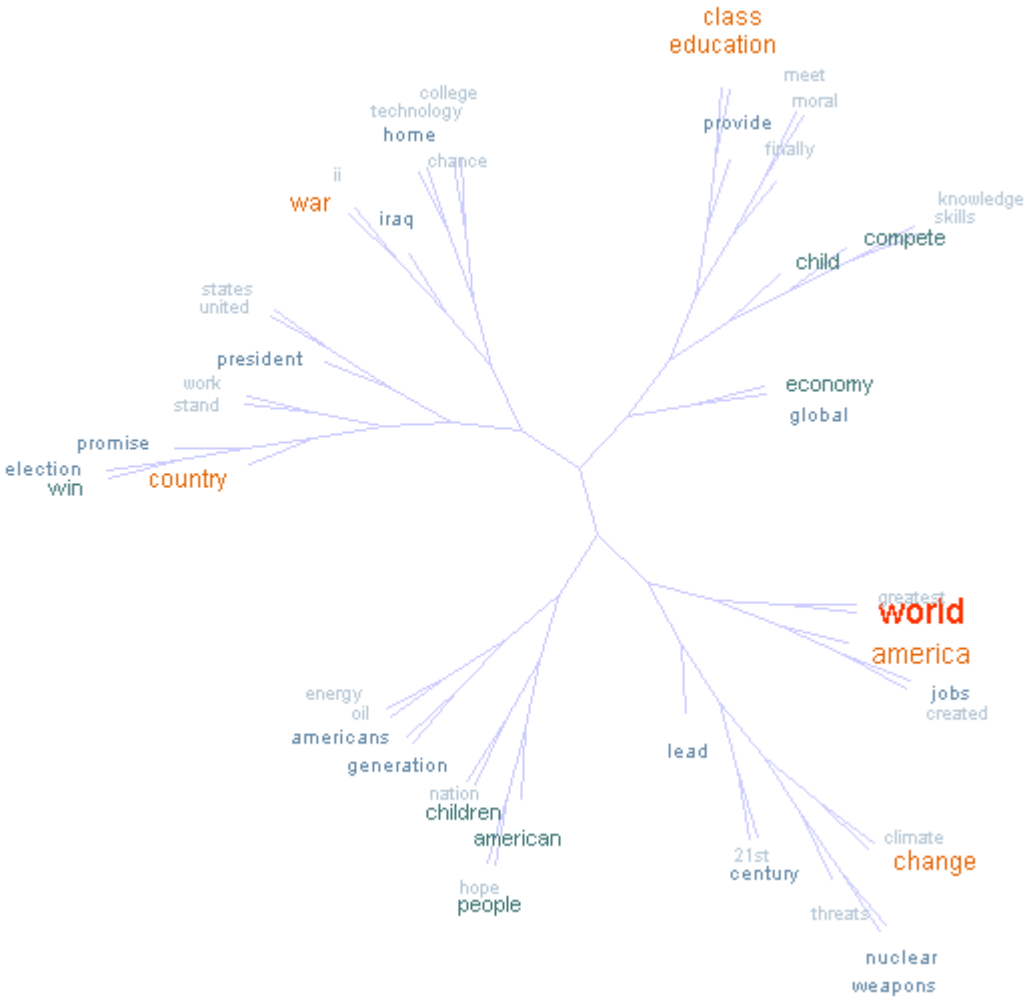
Tree clouds focused on one word



Tree cloud of the neighborhood of "Bush" in Obama's campaign speeches

http://www.treecloud.org - http://www.splitstree.org

Tree clouds focused on one word



Tree cloud of the neighborhood of "world" in Obama's campaign speeches

http://www.treecloud.org - http://www.splitstree.org